Additional Experiments

I. Perception

A. Multiple Marker Identification

In setups involving multiple markers, each marker can be assigned a distinct flashing frequency, allowing the DVS to distinguish them by a simple threshold-based frequency filter, where the maximum event rate in each RoI is computed and compared. As shown in Fig. 1, the left marker is 500Hz while the right marker is 200Hz.

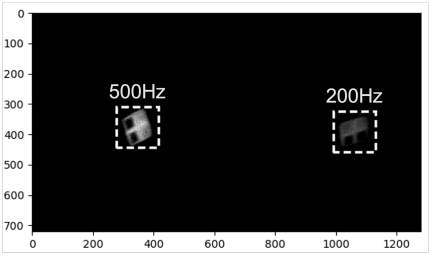
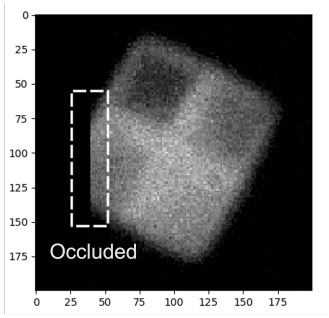


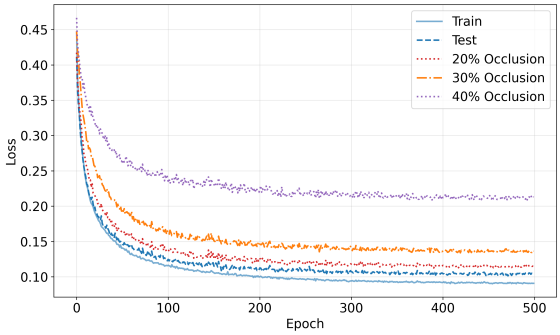
Fig. 1. Integral map of two markers with different frequency in the same frame.

B. Occlusion Scenario

We have also tested the performance of our algorithm on occluded dataset, as shown in Fig. 2 (a), where the integral map of a left-occluded marker is illustrated. We test the performance on dataset of 20%, 30%, 40% occlusion percentage. As shown in Fig. 2 (b), the accuracy drops drastically when occlusion percentage exceed 30%.



(a)



(b)

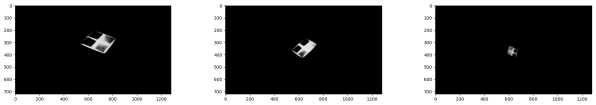
Fig. 2. Occlusion scenario analysis. (a) An example integral map of a marker with 20% left occlusion. (b) Pose estimation error on occluded dataset.

C. Ambient Flicker Disturbance

Figure 3 (a) and (b) shows the visualization of our marker in the presence of an ambient flicker source (a lamp) while the RM moves toward the marker at 0.17m/s. As illustrated, the low-frequency flicker noise is effectively suppressed because its temporal characteristics differ significantly from the high-frequency modulation of our LED marker. The frequency separation allows reliable marker detection under ambient flickering conditions.

D. Distance Estimation Generalizability

The quadratic regression for distance estimation was implemented using data collected at five known distances (40, 50, 60, 70, and 80 cm). The fitting process follows a standard approach implemented via numpy.polyfit. This process is summarized in Table I, which reports the average pose estimation and distance regression errors at 25 cm, 45 cm, and 95 cm, each averaged over 100 samples. Integral map of example sample is shown in Fig. 4



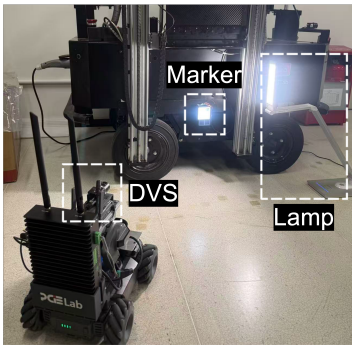
(a) (b) (c)

Fig. 4. Example integral map at different unseen distances. (a) 25cm; (b) 45cm; (c) 95cm.

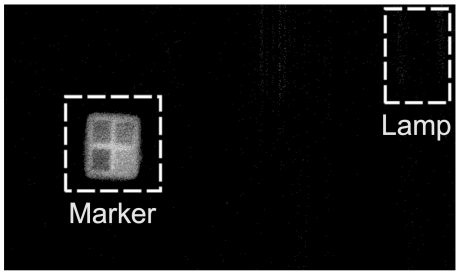
TABLE I
Distance Estimation Error in Unseen Scenarios

	25cm	45cm	95cm
$L_a(^\circ)$	3.82	2.95	4.01
$L_d(cm)$	0.6	0.3	1.9

As shown in the results, our proposed method achieves decent performance at 45 cm, as this range is well covered in the training data. At 25 cm, the marker appears too close to the DVS, causing distortion in the captured pattern and thereby degrading performance. At 95 cm, the marker becomes too small, offering limited spatial resolution, which also affects the estimation accuracy.



(a)



(b)

Fig. 3. (a) Scenario with a lamp as ambient flicker. (b) Marker perceived by DVS with a duration of 2ms with ambient flicker aside.

II. Control

A. Control Gains and Noise Scale

To analyze the relationship between the control gains and observation noise scale, we change the noise scale in the simulation, where the noises scale of coordinate measurements are set to $\Delta_{\text{noise}} = 0.1m$ and $0.25m$. First, the corresponding trajectories for various noise scales and the values of the derivative gain $\boldsymbol{K}_d$ are shown in Fig. 5. Specifically, when $\boldsymbol{K}_d = \text{diag}(0, 0)$, the system exhibits chattering behavior and fails to reach the desired waypoints within the given time, which shows the importance of the derivative term for damping and stability; When the observation noise is increased to 0.25 m, the control task also fails if $\boldsymbol{K}_d$ remains at the same value, and it needs to be decreased accordingly to complete the task. As shown in Fig. 5, with appropriate selection of $\boldsymbol{K}_d$, the task can be successfully completed even under a larger noise scale of 0.25, although slight chattering may occur due to the noise. Second, when the noise scale exceeds a certain threshold, we need to slightly decrease $\boldsymbol{K}_1$ and increase $\boldsymbol{K}_2$. With this tuning, the FFC term provides most of the control effort, while the ST-SMC term primarily compensates for residual disturbances without being overly sensitive to the Gaussian noise. The aforementioned has been clarified on page 17-18 of this revised version.

B. Transfer Learning

For transfer learning, we manually operated the vehicle under various conditions while carrying different payloads like

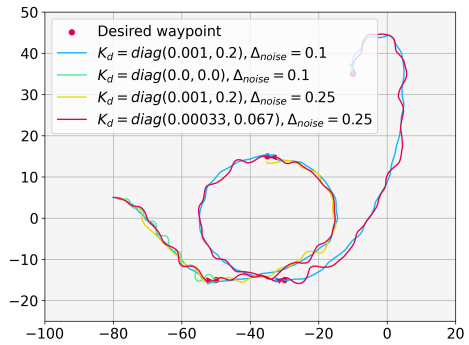
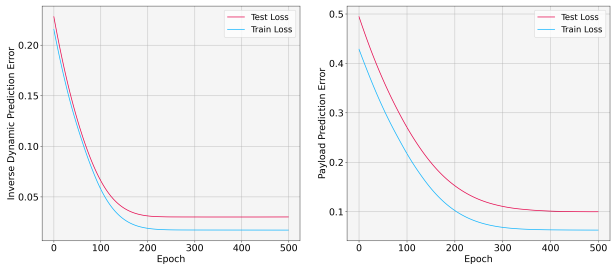

Desired waypoint
$K_d = diag(0.001, 0.2), \Delta_{noise} = 0.1$
$K_d = diag(0.0, 0.0), \Delta_{noise} = 0.1$
$K_d = diag(0.001, 0.2), \Delta_{noise} = 0.25$
$K_d = diag(0.00033, 0.067), \Delta_{noise} = 0.25$

Fig. 5. Performance comparison of different $\boldsymbol{K}_d$ with different observation noise Δ_{noise}.

0kg, 10kg, 20kg, 30kg. During these trials, we performed a diverse set of maneuvers including acceleration, deceleration, sine-wave turns, and several complete inspection routes, to collect transition data for training. This dataset of a total of 10,617 transitions captured the basic dynamic responses of the vehicle under varying loads in realistic scenarios. It is important to note that overfitting must be carefully avoided, as the amount of real-world data was significantly smaller compared to the abundant data available in simulation. To address this, we employed strategies such as limiting model complexity and freezing pretrained parameters to prevent overfitting. The transfer learning process of IDM and PE is shown in Fig. 6



(a) (b)

Fig. 6. Transfer learning process, where the data split ratio is 8:2, and all data have been normalized. (a) IDM (b) PE.

C. Payload Placement Effect

To analyze the effect of payload placement on the vehicle's overall dynamics, we conducted a simulation in which the throttle and steering torque inputs were kept identical across three scenarios: 1) a 20 kg payload placed at the front of the vehicle, 2) a 10 kg payload placed at the front, and 3) a 20 kg payload placed at the rear. The resulting controlled trajectories are shown in Fig. 7. The figure shows that payload placement is just as important as payload weight in influencing the vehicle's overall dynamics.

D. PE Time Window

To analyze the relationship between PE identifiability and the window length n_w, the vehicle with a 20 kg payload was

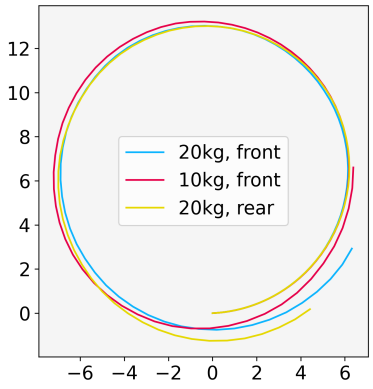


Fig. 7. Trajectories of the vehicle under identical control inputs with varying payload weights and placement.

driven under a fixed control input, and the resulting trajectory was processed by the PE module using different window lengths, as shown in Fig. 8. The results indicate that a smaller n_w yields lower prediction latency initially, but leads to more chattering in the prediction compared with a larger n_w. To balance responsiveness and smoothness, we selected a mid-range value of $n_w = 20$.

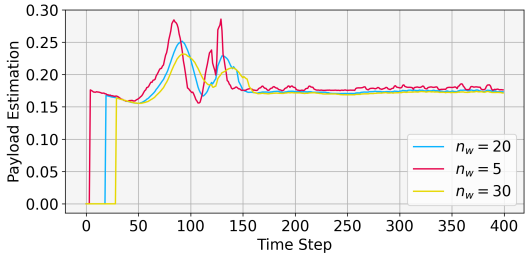


Fig. 8. Estimated payload with different n_w.